FPGAでソフト・マクロのCPUを使う理由

“FPGAマイコン”の特徴と将来への期待

浅井 剛

2000年にFPGAベンダが自社製品向けにCPUコアをリリースしてから7年が経過した．ここでは，FPGA向けに提供されているソフト・マクロのCPUを活用する理由について解説する．特に，一つのLSIに複数のCPUコアを搭載するマルチコアに注目する．少量多品種の開発でCPU処理を行いたい場合，32ビットCPUに周辺回路をあらかじめ搭載した汎用マイコンを使うことが多い．しかし汎用マイコンとリアルタイムOSの組み合わせでは，高まる要求仕様を満たせない場合がある．高い性能を得るために動作周波数を上げると，消費電力が増えてしまうといった問題がある．このような状況を解決してくれるのがFPGAでCPUコアを活用する“FPGAマイコン”である．FPGAの大規模化に伴い，一つのFPGAで複数のCPUコアを活用することも現実的になっている．（編集部）

汎用マイコンなどでは実現困難な機能を達成するために，カスタムLSIは不可欠です．しかし少量多品種の開発では，ASIC（Application Specific Integrated Circuit）の開発を断念せざるを得ません．FPGA（Field Programmable Gate Array）とソフト・マクロのCPUの組み合わせは，独自機能のSOC（System on a Chip）を得るための，唯一の選択肢です．

1．汎用マイコン＋リアルタイムOSの限界とマルチコア

組み込みシステムといえば，16ビットか32ビットの汎用マイコンが機能の中心になるのが一般的です．最近では，システムLSI（SOC）として1チップ化することも可能になっています．ソフトウェアの開発効率を上げるために，リアルタイムOSも広く利用されています．この構成はコスト・パフォーマンスにも優れています．

● 一つのCPUですべて処理しようとするから困難になる

汎用マイコンを用いた組み込みシステムの開発過程では，「タスクが多くなりすぎてシステム全体のデバッグが難しい」とか，「割り込み要因が多くなりワーストケースでアプリケーションの処理時間を満たせない」という課題にぶつかることが珍しくありません．前者についてはタスク分割の見直しや優先順位を再整理することで，後者については割り込みハンドラのチューニングなどで対応することになります．

しかし，どうしてもソフトウェアの組み替えだけでは対処しきれないこともあります．CPUの動作周波数を上げられればよいのですが，できない場合は製品としての機能・性能を下げるか，開発初期段階まで戻るかの決断を迫られます．

そのような経験を何度となくしていると，ある疑問が湧いてきます．

「なぜ一つのCPUですべてを処理しなければならないのだろうか？」

具体的な事例を以下に示します．

● 事例1：周辺機能側でデータ前処理を行ってしまう

32ビットCPUにリアルタイムOSを搭載したシステムの

KeyWord　FPGA，ソフト・マクロ，CPUコア，マルチコア，汎用マイコン，SOC，リアルタイムOS，マルチプロセッサ，負荷分散型，機能分散型

例を図1に示します．

このシステムでは，アプリケーション・タスクで32ビットのデータ処理性能が求められているとします．割り込みハンドラでは，リアルタイムOSの周期タイマや外部とのシリアル通信（SCI：Serial Communication Interface），スイッチ入力などを処理します．

ここでシリアル通信による入力を考えてみましょう．SCI割り込みハンドラとして必要な機能は，

①SCI受信バッファから受信文字の読み出し
②受信可能文字かの判定
③1文字のエコーバック（受信不可能文字の場合はエラー・コード）出力
④受信バッファへの格納
⑤通信終了デリミタ検出
⑥受信完了のイベント・フラグを立てる

とします．

本システムにおいて，8ビット（1文字）単位で行われるこれらの処理を32ビットCPUで実行する必要があるのでしょうか．必要性だけを考えれば誰が見ても答えは“NO”です．しかし，システムのCPUが32ビットで，かつ1個しかないため，やむなく処理させているだけです．処理性能に余裕があり，その余力の範囲内で対応可能であれば何の問題もないかもしれません．しかし，例えばシリアル通信が高速で，頻繁に割り込みを発生させる場合，本来処理しなければならないアプリケーション・タスクに対して大きなオーバヘッドとなります．
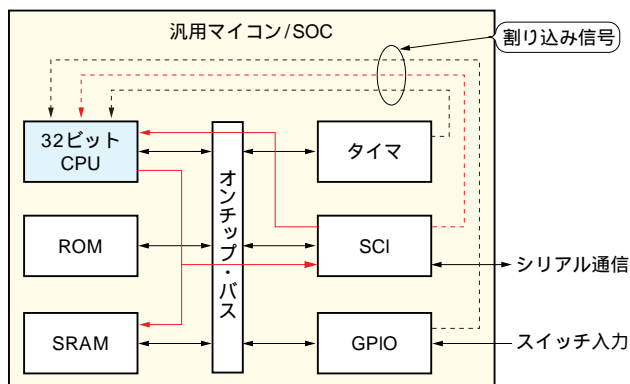
図1　32ビットCPUを搭載し，リアルタイムOSを搭載したシステムの例

SCI受信割り込みハンドラのデータの流れを示す．CPUでは，SCI受信バッファから受信文字を読み出し，受信可能文字か判定し，1文字のエコーバック（受信不可能文字の場合はエラー・コード），受信バッファへの格納，通信終了デリミタ検出，受信完了のイベント・フラグを立てるといった処理を行う必要がある．水色はバス・マスタ機能を有しているモジュールを表す．

そこで，もしSCI側で通信処理を行い，かつバス・マスタとしてSRAMに直接受信データを書き込める機能があったらどうでしょうか（図2）．このSCIに求められるデータのビット幅は8ビットですから，SCIにCPUを内蔵するとしても8ビットで済むことになります（SRAMへのフル・アドレス生成機能は必要）．

この場合32ビットCPUがSCIからの割り込みによって行う処理は，通信受信完了のイベント・フラグを立てるだけになります．OSの内部状態を変えるこの処理だけは必要ですが，言い換えればSCIにシーケンス処理＋バス・マスタ機能［こちらはDMA（Direct Memory Access）コントローラで実現してもよい］を追加するだけで，CPUに対して最少のオーバヘッドを実現できるわけです．

● 事例2：二つのCPUに機能を分散して処理する

一定周期（数百μs）で割り込みによってA-Dコンバータからアナログ情報を読み出し，所定の処理後PWM（Pulse Width Modulation）制御をかけるシステムの例を図3に示します．システム全体のシーケンス制御は数十ms単位でよく，高精度な処理も必要ありません．

CPUが一つの場合，数百μsごとの32ビット処理の余力（10％～20％）を使ってリアルタイムOS＋シーケンス制御のアプリケーション・タスクを実行させることになります．このため，PWM制御が重くなるにつれてシーケンスの応答性確保に影響が出ることになります．

その解決策が図4です．高精度でかつ高速な応答が求められる部分を32ビットCPUの処理系で構成します．決められた処理だけで，余計なシーケンス処理を担当しないの
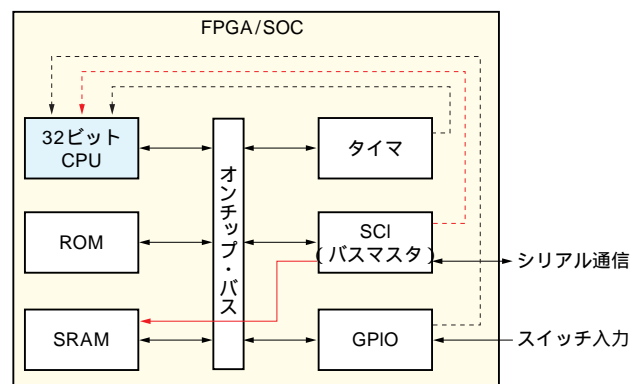
図2　SCI部でデータ処理を行い，バス・マスタ機能を備える

SCIからの割り込みによって行う処理は，通信受信完了のイベント・フラグを立てるだけになる．水色はバス・マスタ機能を有しているモジュールを表す．

で，リアルタイムOSは必要ありません．従って，制御応答の高速化も可能です．一方，数十ms周期でよいシーケンス制御では，8ビットもしくは16ビットのCPUを使います．リアルタイムOSを利用する必要はあるかもしれません．しかし短い周期の割り込みがなくなることから，余裕を持った処理が可能です．二つのCPUは非同期でよく，必要に応じて共有メモリか32ビットCPUからの割り込みをかけて，通信を行います．

● CPUを必要なだけ搭載するのが理想的なシステム

この二つの事例では，一つのCPUを複数の目的で使用したために，本来の処理性能に影響を与えます．そしてその対策には，マルチバス・マスタ（**図2**）やマルチプロセッサ・システム（**図4**）が最適解であることを理解いただけたと思います．

汎用マイコンを使ったシステムや，すでに設計を終えてしまったASIC（SOC）では，CPUの数を簡単に変更できません．ところがFPGAは，リソースが許す範囲であれば，ソフト・マクロのCPUを必要に応じて追加可能です．

FPGAとソフト・マクロのCPUによる組み合わせは，マルチプロセッサ・システムという最適解を得るために，一番身近な手法なのです．

2. マルチプロセッサ・システムの構成方式

FPGAでマルチプロセッサが簡単に実現できるといっても，実際の搭載にあたってはさまざまな知識や技術が必要になります．ここでは負荷分散型（対称型）と機能分散型（非対称型）と呼ばれる一般的なマルチプロセッサ・システムの構成について説明します．

● 負荷分散型（対称型）

負荷分散型（対称型）は同じCPUを複数個搭載し，実行するタスク（ソフトウェア）を動的に制御する方式です．単独のCPUでは動作周波数で処理性能が決まります．複数のCPUを搭載すれば，動作周波数を上げることなく総合的な処理性能を向上させられます．パソコン向けCPUでは，もはや当たり前になってきた2コアや4コアは，この負荷分散型に相当します．

実行するタスクの動的な制御はOSなどで行います．見かけ上一つのCPUのように振る舞うので，ユーザはあるソフトウェアがどのCPUで実行されるかを意識する必要がありません．

ただし，CPUごとにキャッシュを持っていた場合，そのコヒーレンシ確保に高度なしくみが必要になります．このため，サーバのような高価なシステム向けのアーキテクチャであり，どちらかというと組み込み機器向けではありません．

図3　アナログ情報を処理しPWM制御をかけるシステムの例
一定周期（数百μs）で割り込みごとに行う高精度の演算が処理の中心である．割り込み処理の余力で，アプリケーション・タスクを実行する．
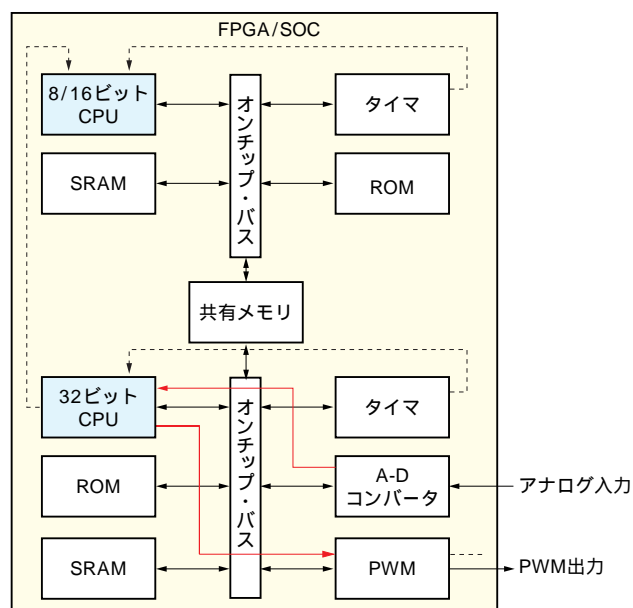
図4　シーケンス処理と割り込み処理のCPUを分ける
32ビットCPUは決められた処理だけしか行わないのでリアルタイムOSは必要なく，制御応答の高速化も可能になる．数十ms周期でよいシーケンス制御では，8ビットもしくは16ビットのCPUを使う．

● 機能分散型(非対称型)

機能分散型(対非称型)は，搭載したCPUごとにあらかじめ処理すべき機能を分割する方式です．CPU間は通信手段によって分離されているので，同じアーキテクチャのCPUである必要もなく，対称型に比べ非常に自由度の高い構成が可能です．求められた機能をその処理に最適なCPUを割り当てて処理させます．1チップ/マルチチップという実装方法に関係なく，まさに組み込み機器向けのアーキテクチャです．

各CPUを一つの機能(システム)と位置付け，それぞれに異なるOSを搭載(OSレスの場合もあり)することもあります．この機能分散型マルチプロセッサ用リアルタイムOSとして，TOPPERSプロジェクトのTOPPERS/FDMP(Function Distributed Multiprocessor)カーネルがあります．2005年にカーネル仕様が公開され，2006年4月にオープン・ソースとして公開されました．そして本稿執筆時点では，英国ARM社の「MPCore」，米国Altera社の「Nios II」，米国Xilinx社の「MicroBlaze」，東芝の「MeP」がサポートされています．TOPPERSプロジェクトのホームページ(http://toppers.jp/fdmp-kernel.html)からバージョン1.1.2のカーネルのソース・ファイルがダウンロードできます．

3．“FPGAマイコン”への期待

これまではFPGAにとって良いことばかりを書いてきました．しかしFPGAは，機能面でASICと勝負できない面もあります．また，CPUアーキテクチャにディスコン(生産中止)がないという点にメリットを感じてソフト・マクロのCPUを採用したFPGA設計者は，FPGAベンダに対して今後の更なる展開を期待しています．

● ASICのように最適なパッケージを選択したい

FPGAは，プログラマブル・デバイスという特徴から，論理規模が増えるに従ってユーザI/O数も増える傾向があります．このため，最近では最大で1,760ピンのパッケージが使われています．一方，同等の回路規模のASICで求められるのは，アプリケーションにもよりますが400～600ピン程度です．

超多ピン・パッケージは，パッケージ・コストだけでなく，実装面積でも不利になります．汎用マイコンやSOCの代わりにFPGAとCPUコアを活用したいと考えても，この超多ピン・パッケージしか使えないのでは勝負できません．

汎用性を極めてきたFPGAに対し，特定のアプリケーションを想定してパッケージを用意してほしいというのは難しいのかもしれません．I/O規格への対応を限定したりピン数を減らしたパッケージ，さらに欲を言えばCPUコアを活用するためにできる限り多くのメモリを搭載した製品があると，さらに使いやすくなると思います．

● 無償ツールとの組み合わせもサポートしてほしい

FPGAベンダは，無償版の設計ツールを提供しています．有償版に比べて機能などに制限がありますが，ターゲット・デバイスが無償版でサポートされている場合には，無償版が製品開発に使われることも多いようです．

筆者の知人が最近CPUコアのライセンスを購入しようとしたら，正規版のツール購入が前提と言われたと連絡して来ました．FPGAベンダのホームページを見てみると，確かに注意書きとして明記されています．さらに，以前あったCPUコア専用のライセンス申請口もなくなっています．

有償版のライセンス申請時に使用するドングルやネットワーク・インターフェース(MACアドレス)単位にユーザ

コラム　8ビットと32ビットの互換マイコン

米国Freescale Semiconductor社の「Flexisシリーズ」は，ピン互換性があり，同一の周辺機能を持ち，CPUコアだけが異なる8ビットと32ビットのマイクロコントローラです．一つのボードとソフトウェア・コード，開発ツールで，ローエンドからハイエンドまでの組み込みシステムをサポートしてしまおうというコンセプトの製品です．もちろんCPUコア自体は異なるので，再コンパイルは必要です．共用するためにはコードの記述に注意が求められますが，それだけで移行できてしまうという優れものです．

コストと消費電力では優っても，ハードウェアの柔軟性ではFPGAに絶対勝てなかった汎用マイコンがこのような形で柔軟性を備えたことに驚きました．

リコンフィギャラブル・プロセッサを筆頭に，ハードウェアに柔軟性を持たせるさまざまなしくみも提案されています．FPGAも，新しい展開を考えてほしいものです．
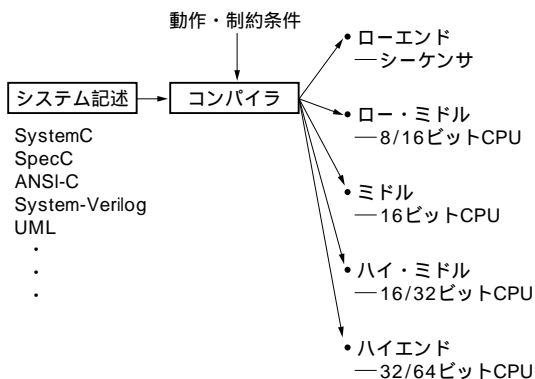
図5　高位合成ツールへの期待

処理単位に書かれたシステム記述に，それを動作させるための制約条件を与えると，単純なシーケンサから多ビットのハイエンドCPUまで生成できる．

を管理し，設計ツールのライセンスに合わせてIP（Intellectual Property）コアのライセンス発行をするシステムになっているためでした．確かにCPUコアもIPコアの一つですから，管理方法を統一したいというのは分かります．しかし，設計ツールによってIPコアの利用が制限を受けるというのは残念なことです．

現在のところ，最もリーズナブルにCPUコアを使った開発を行うためには，FPGAベンダの提供する開発キットを購入することが近道のようです．

無償ツールとCPUコアの正規ライセンスの組み合わせが早期にサポートされることを強く希望します．

● 適材適所のCPUを実現するために高位合成技術に期待

FPGAとソフト・マクロのCPUの組み合わせがマルチプロセッサ化に向いていることは理解していただけたと思います．しかし現在提供されているCPUコアは32ビットRISCプロセッサが中心で，筆者が考えている自由なシステム構成は実現できません．

高位合成技術も実用的になりつつある昨今，ユーザの一人として適材適所のCPUを実現するために**図5**のようになることを期待しています．

処理単位に書かれたシステム記述に，それを動作させるための制約条件を与えると，単純なシーケンサから多ビットのハイエンドCPUまで生成できるというものです．そして，それらを自由に組み合わせてFPGAでシステムを構築します．

このような設計の柔軟性こそ，FPGAがさらに発展していくために必要ではないかと感じています．

あさい・たけし
東京計器工業(株)　技術部
特定非営利活動法人　FPGAコンソーシアム監事

FPGAにおける設計技術セミナー＆展示会
第10回 6都市 **FPGAカンファレンス2007**

本誌筆者による公演のご案内

6都市で開催される「FPGAカンファレンス」（主催：FPGAソーシアム）において，本誌筆者によるFPGA活用に関する講演が開催されます．参加は無料です．

詳細は http://www.fpga.or.jp/

日付	会場
9月14日（金）	東京FPGAカンファレンス　キャンパスイノベーションセンター
10月12日（金）	仙台FPGAカンファレンス　仙台市情報産業プラザ
10月19日（金）	札幌FPGAカンファレンス　ASTYホール（ASTY45ビル4階）
11月7日（水）	名古屋FPGAカンファレンス　名古屋市中小企業振興会館（吹上ホール会議室）
11月30日（金）	なにわFPGAカンファレンス　梅田センタービル
12月7日（金）	博多FPGAカンファレンス　アクロス福岡

全都市　**「FPGA最前線」～FPGAデバイスを取り巻くトピックと新たな活用法～（熊本大学大学院教授　末吉 敏則氏）**

FPGAとは，ハードウェアでありながらソフトウェアのように書き換え変更ができる再構成可能な論理デバイスである．最近のFPGAでは，課題となってきた低価格化も加速度的に進み，イニシャル・コストの低さとも併せて新たな市場への浸透を加速させている．また，リコンフィギャラブル・システムなど，ASICにない再構成可能という特性を活かした利用法にも期待が寄せられている．一方，FPGAの集積度や動作周波数が上がるに伴って消費電力が急増し，FPGAの泣き所ともなっている．ここでは，これら最新FPGAデバイスを取り巻くトピックや技術トレンドについて解説するとともに，新たな活用法について紹介する．

全都市　**「マルチプロセッサ・システムにおけるFPGA活用事例」～市販マイコンからソフトコアCPUまで～（東京計器工業　浅井 剛氏）**

近年益々高度化・複雑化する組み込み機器において，マルチプロセッサを採用するケースが増えている．しかしシステム構成の複雑化に伴い，開発の長期化や出荷後の不具合発生などの問題が発生している．本講演では，市販マイコンからソフト・コアCPUまで幅広い事例を元に，マルチプロセッサ・システムでFPGAの柔軟性をどの様に活用していけばよいかを紹介する．